Abstract

Computer systems are typically designed with multiple levels of memory hierarchy. Prefetching has been employed to overcome the latency of fetching data or instructions from or to memory. Prefetching works well for data structures with regular memory access patterns, but less so for data structures such as trees, hash tables, and other structures in which the datum that will be used is not known a priori. In modern transaction processing systems, database servers, operating systems, and other commercial and engineering applications, information is frequently organized in trees, graphs, and linked lists. Lack of spatial locality results in a high probability that a miss will be incurred at each cache in the memory hierarchy. Each cache miss causes the processor to stall while the referenced value is fetched from lower levels of the memory hierarchy. Because this is likely to be the case for a significant fraction of the nodes traversed in the data structure, processor utilization suffers. The inability to compute the address of the next address to be referenced makes prefetching difficult in such applications. The invention allows compilers and/or programmers to restructure data structures and traversals so that pointers are dereferenced in a pipelined manner, thereby making it possible to schedule prefetch operations in a consistent fashion. The present invention significantly increases the cache hit rates of many important data structure traversals, and thereby the potential throughput of the computer system and application in which it is employed. For data structure traversals in which the traversal path may be predetermined, a transformation is performed on the data structure that permits references to nodes that will be traversed in the future be computed sufficiently far in advance to prefetch the data into cache.